A Project Report

On

“Used Car Price Prediction”

TECHNEX’21

Submitted To:

Mr. Mayur Dev Sewak
General Manager, Operations
Eisystems Services

Ms. Mallika Srivastava
Trainer, Data Science & Analytics Domain
Eisystems Services

Submitted By:

Rushik Patel

Index

1. List of Figures

2. Abstract

In this report, we look at how supervised machine learning technique can be used to forecast used car prices in India. Data from online website kaggle was used to make the predictions. The prediction was made using multiple linear regression analysis. The predictions are then analysed and compared to determine how much accuracy does the algorithm provides. A seemingly simple problem turned out to be difficult, as dataset has to be thoroughly clean for better accuracy. In future, we can use more advance algorithm to predict car prices with increase in accuracy.

3. Summary

The Price of new car is fixed by manufacturer with additional taxes of state as well as central government. Therefore, customer knows exact amount of money for buying new car. But, due to increase prices of new car and financial incapability of customer to buy them, second hand car market has been growing in popularity. These results in providing opportunity for both buyers and sellers. Buying the used car is best option for customer because the price is fair and affordable.

However, many factors affect the price of used car, including its age, current condition, mileage, engine efficiency and many more. In most of cases, the price of used car on the market fluctuates. As a result, model for evaluating car price is needed.

To overcome this, I developed a model based on multiple linear regression that can predict car price. Predicting used car's resale value is not an easy job as it depends on number of other variables. The most significant are car's age, model, engine, power (bhp) and mileage(kmpl). As this model produces continuous value as an output and not categorical, giving all these variable as input model predicts actual price of car.

4. Objectives

Deciding whether a used car is worth the posted price when listed online can be difficult. Several factors like mileage, year, model, etc. can influence car price. From seller's perspective, it is also a dilemma to price a used car appropriately. Based on existing data, the goal is to use machine learning algorithm for developing model to predict used car price.

In addition, to consider each effective factors while predicting car price. Therefore, to develop an efficient and effective model which predicts the price of a used car according to user's inputs with good accuracy.

5. System Requirements

Hardware Requirements

- ✓ Operating System - Windows 7, 8, 10
- ✓ Processor - dual core 2.4 GHz (i3, i5 series intel processor or equivalent other processor)
- ✓ RAM – 4 GB

Software Requirement

- ✓ Python
- ✓ Jupyter Notebook (recommended) or Pycharm
- ✓ Chrome (web browser)
- ✓ PIP 2.7

6. Data Flow Diagram/ Algorithm

The project starts with collecting dataset. The next step is to do Data cleaning, Data Preprocessing and feature selection. Data Preprocessing includes Data reduction, Data Transformation. Then, using machine learning algorithm we will predict the price. The algorithm used is multiple Linear Regression. Then model predicts the most accurate price. After that predicted price is displayed to the user according to user's inputs.

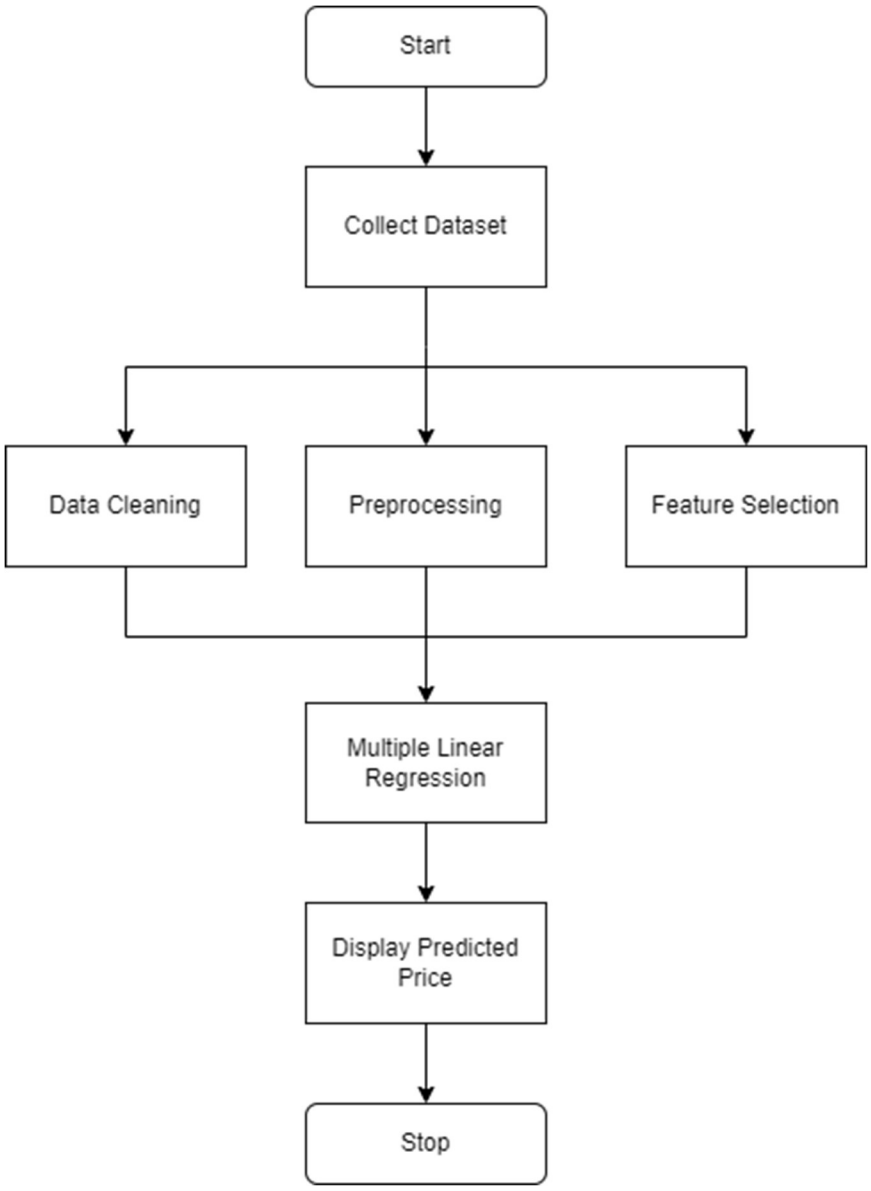

Fig 5.1 Flowchart of Project

6.1 Dataset

For this project, I have used dataset on used car sales available on kaggle. The features available in this dataset are Name, Location, Year, Kilometers_Driven, Fuel_Type, Transmission, Owner_Type, Mileage (kmpl), Engine (CC), Power (Bhp), Seats, New_Price and Price.

A Short glimpse of first five row and last five rows of dataset is shown below

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74

Fig 6.1.1 Head of Dataset

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	NaN
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	First	17.21 kmpl	1197 CC	103.6 bhp	5.0	NaN	NaN
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	NaN
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	Third	17.2 kmpl	1197 CC	103.6 bhp	5.0	NaN	NaN
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	First	10.0 kmpl	2148 CC	170 bhp	5.0	NaN	NaN

Fig 6.1.2 Tail of Dataset

The link for the dataset is https://www.kaggle.com/yogidsba/predict-used-car-prices-linearregression/data.

6.2 Data Cleaning, Preprocessing and Feature Selection

Data Cleaning

In order to get better understanding of data, I first dropped all the data having null values. Then I converted some features into float values for processing. For example Mileage, Engine and Power where trailed by kmpl, CC and Bhp respectively that makes them object datatype. After converting them into float values they are ready for further processing.

Data Preprocessing

After cleaning data and making, them ready to visualize. Data visualization is the best way to find out how data looks like. I plotted various graph to get overview of our target column that is price vs various other column.

1. First I created histogram of car prices. This figure shows how car prices are distributed over the data. For example, majority of data is distributed between 0-20.

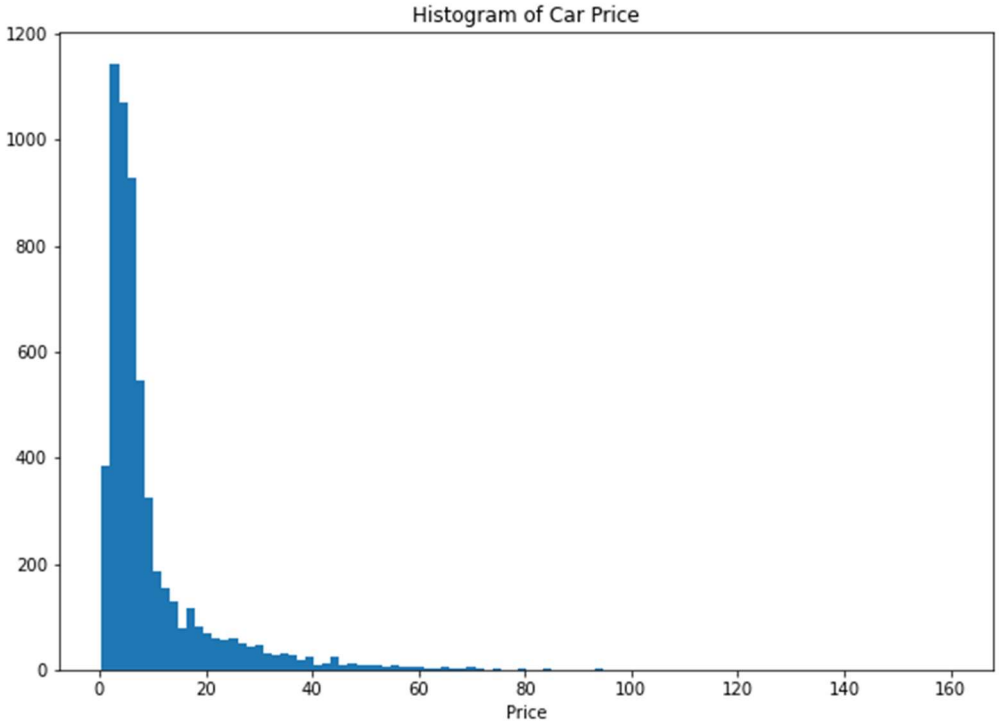

Fig 6.2.1 Car Price Histogram

2. Then I plotted box plot of Fuel type vs Price to get insight on which type of vehicle is priced more and which is priced less.

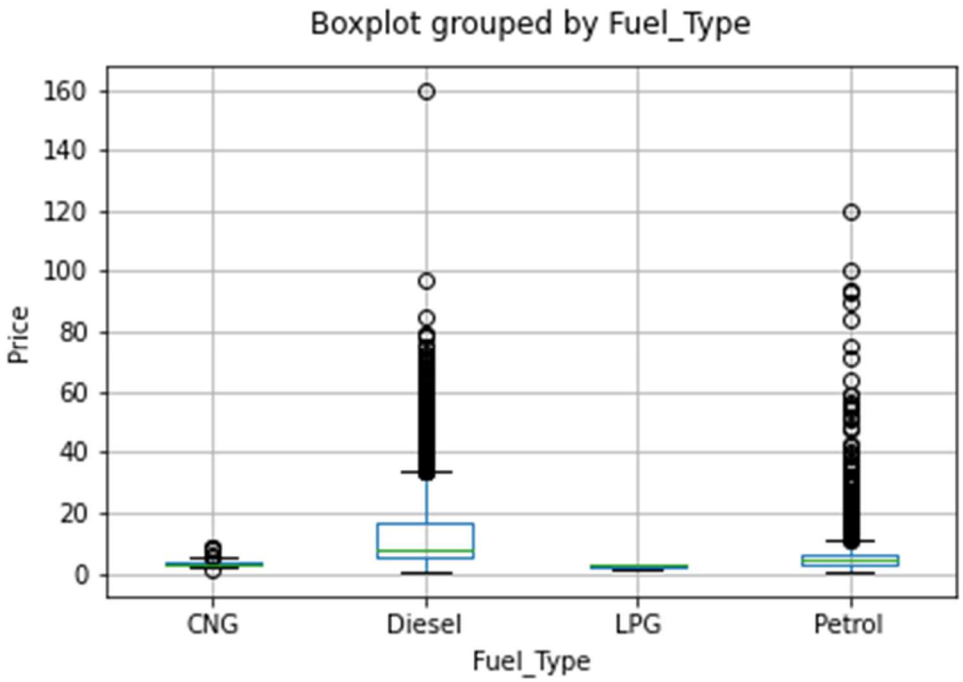

Fig 6.2.2 Fuel_Type vs Price

Above is the box plotted figure of Fuel type vs Price. As, we thought diesel car would cost followed Petrol and then CNG and LPG.

3. After Fuel_Type vs Price boxplot I plotted box plot of Owner type vs Price. This helped me to gain knowledge about which car owner is willing to pay more for the car.

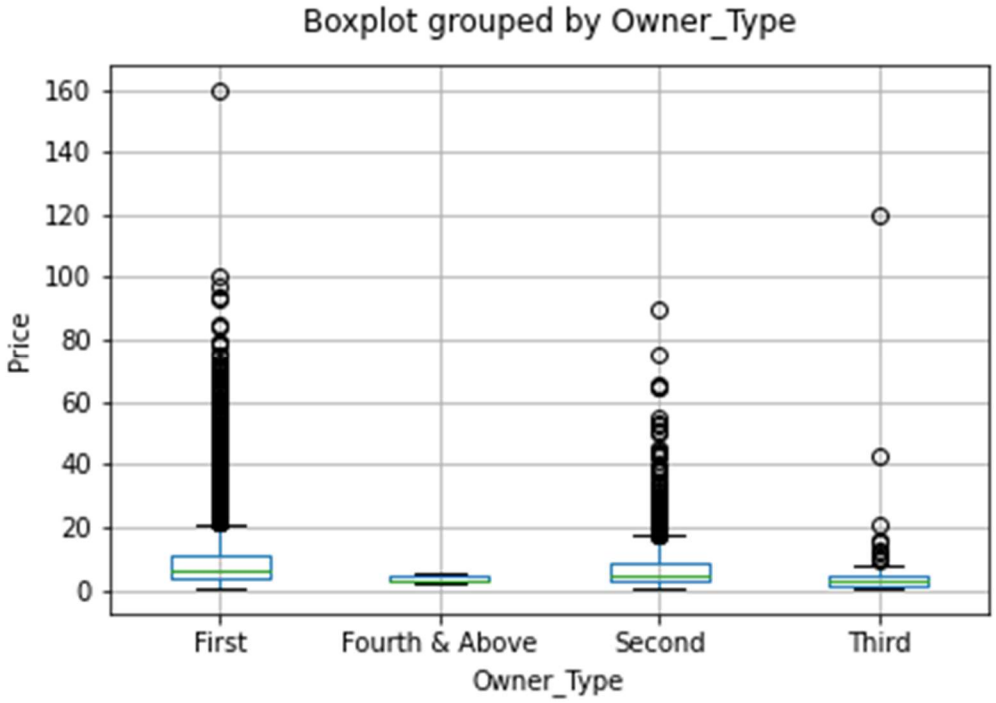

Fig 6.2.3 Owner_Type vs Price

As we see in figure below first hand owner is willing to pay more followed by second hand and then third and fourth and above.

4. At last, I plotted bar graph of company vs its products count. This graph shows which car company is common and is more in dataset compared to other car company. For example, Maruti is most common brand followed by Hyundai.

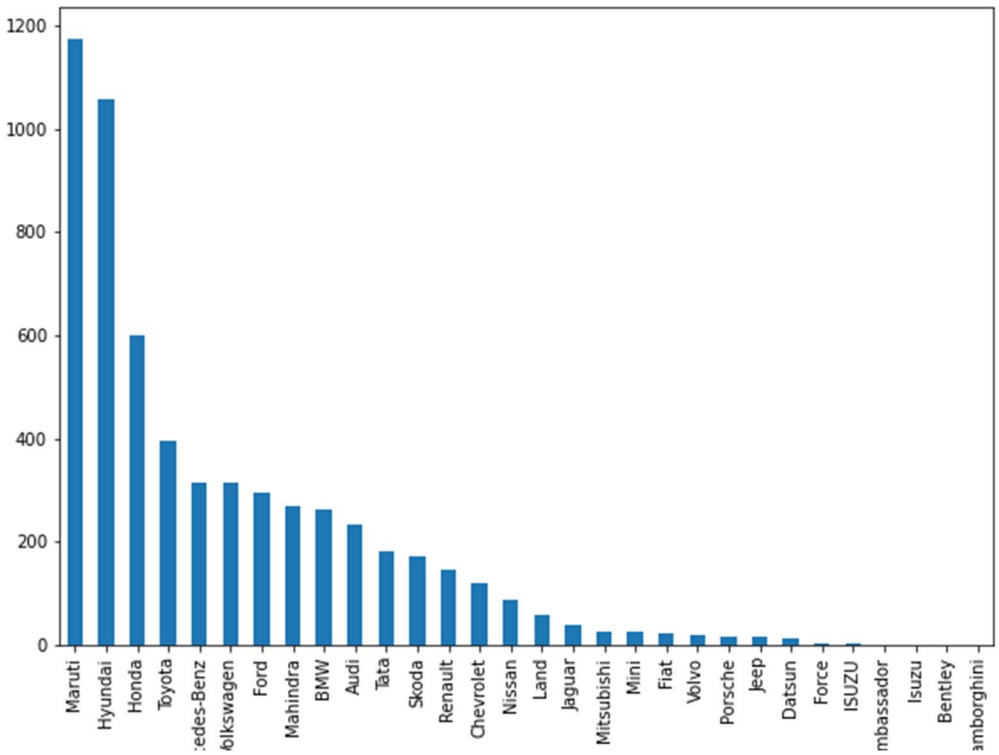

Fig 6.2.4 Company vs Count

Feature Selection

Now it is more important to select appropriate feature from the dataset that would be input to the model which predicts output. We convert some categorical features into categories such as location, Fuel_Type, Transmission which is used as an input into the model.

6.3 Multiple Linear Regression

Linear Regression attempts to model the relationship between two variables by fitting a linear equation to observed data. It performs the task to predict a dependent variable value (y) based on a given independent variable(x).

Similarly, multiple linear regression is a statistical technique that is used to analyse the relationship between a single dependent variable and several independent variable. The objective of multiple regression analysis is to use the independent variables whose values are known to predict the value of the single dependent variable. Each predictor value is weighted, the weights denoting their relative contribution to the overall prediction.

$$Y = a + b_1 X_1 + b_2 X_3 + \ldots + b_n X_n$$

Here Y is the dependent variable, and X1, …. ,Xn are n dependent variables. In calculating the weights, a, b_1,…,b_n, regression analysis ensures maximal prediction of the dependent variable from the set of independent variables. This is usually done by least squares estimation.

6.4 Display predict price

After that, independent variables are given as an input to model. Model built with multiple linear regression predicts output as a dependent variable i.e. in our case it is car price.

7. Code/ Program

```
#  Importing necessary libraries and Dataset
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Importing dataset
df = pd.read_csv('used_cars_data.csv')
df.head()
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74

Fig 7.1 Data Set

```
# finding null values, droping them and reseting index.
print("Shape of df Before dropping any Row: ",df.shape)
df = df[df['Mileage'].notna()]
print("Shape of df After dropping Rows with NULL values in Mileage: ",df.shape)
```

```
df = df[df['Engine'].notna()]
print("Shape of df After dropping Rows with NULL values in Engine : ",df.shape)
df = df[df['Power'].notna()]
print("Shape of df After dropping Rows with NULL values in Power  : ",df.shape)
df = df[df['Seats'].notna()]
print("Shape of df After dropping Rows with NULL values in Seats  : ",df.shape)
df = df[df['Price'].notna()]
print("Shape of df After dropping Rows with NULL values in Seats  : ",df.shape)
```

```
Shape of df Before dropping any Row:  (7253, 13)
Shape of df After dropping Rows with NULL values in Mileage:  (7251, 13)
Shape of df After dropping Rows with NULL values in Engine :  (7205, 13)
Shape of df After dropping Rows with NULL values in Power  :  (7205, 13)
Shape of df After dropping Rows with NULL values in Seats  :  (7198, 13)
Shape of df After dropping Rows with NULL values in Seats  :  (5975, 13)
```

Fig 7.2 Removed Null Values

 # getting company name from name column and converting 'Mileage', 'Engine' and 'Power' column into float datatype.

```
for i in range(df.shape[0]):
    df.at[i, 'Company'] = df['Name'][i].split()[0]
    df.at[i, 'Mileage(kmpl)'] = df['Mileage'][i].split()[0]
    df.at[i, 'Engine(CC)'] = df['Engine'][i].split()[0]
    df.at[i, 'Power(bhp)'] = df['Power'][i].split()[0]

df['Mileage(kmpl)'] = df['Mileage(kmpl)'].astype(float)
df['Engine(CC)'] = df['Engine(CC)'].astype(float)
df['Power(bhp)'] = df['Power(bhp)'].astype(float)
```

Company	Mileage(kmpl)	Engine(CC)	Power(bhp)
Maruti	26.60	998.0	58.16
Hyundai	19.67	1582.0	126.20
Honda	18.20	1199.0	88.70
Maruti	20.77	1248.0	88.76
Audi	15.20	1968.0	140.80

Fig 7.3 Converting Features into float datatype

```
# As for now there are five categorical features.
#1.Location
#2.Fuel_Type
#3.Transmission
#4.Owner_Type
#5.Company
# Dividing these each features into categories and generating new columns.

df=pd.get_dummies(df, columns=['Location', 'Fuel_Type'], drop_first=False)
df=pd.get_dummies(df, columns=['Transmission'], drop_first=True)
df.replace({"First":1,"Second":2,"Third": 3,"Fourth & Above":4},inplace=True)
```

Location_Kochi	Location_Kolkata	Location_Mumbai	Location_Pune	Fuel_Type_CNG	Fuel_Type_Diesel	Fuel_Type_LPG	Fuel_Type_Petrol	Transmission_Manual
0	0	1	0	1	0	0	0	1
0	0	0	1	0	1	0	0	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...
0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	1
0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	1

Fig 7.4 Categorical Data to Features

```
# Feature Selection
# Selecting final features that will be used for model building and droping all other useless feature.

df.drop(["Company"],axis=1,inplace=True)
df.drop(['New_car_Price'],axis=1, inplace=True)
y=df['Price']
df.drop(['Price'],axis=1, inplace=True)
x=df

# model building
# Building model using sklearn library.
# First splitting data to train and test into 80:20 ratio for the model.

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)

# Applying Linear Regression algorithm using sklearn library.
```

```
from sklearn.linear_model import LinearRegression
multi_model = LinearRegression()
multi_model.fit(x_train, y_train)

# Predicting y_test by giving x_test as an input to the model
y_pred=multi_model.predict(x_test)

# Model accuracy for x_train,y_train and x_test,y_test respectively.
multi_model.score(x_train,y_train)
multi_model.score(x_test,y_test)
```

```
y_pred
```

```
array([ 8.37728023, 11.44569692,  9.41778251, ...,  1.99281629,
       13.57967892,  7.1036287 ])
```

```
# Model accuracy for x_train,y_train and x_test,y_test respectively.
```

```
multi_model.score(x_train,y_train)
```

```
0.7015609092235184
```

```
multi_model.score(x_test,y_test)
```

```
0.7270531191250961
```

Fig 7.5 Predicted Output and Accuracy

(For full code/program please refer to following link: https://github.com/Rushik2900/Used-Car-Price-Prediction/blob/main/Car_Price_Prediction_Linear_Reg.ipynb)

8. References

[1] https://www.kaggle.com/yogidsba/predict-used-car-prices-linearregression/data

[2] “Used Car Prediction” by Praful Rane, Deep Pandya, Dhawal Kotak (IRJET 2021)

[3] “Prediction of Car Price using Linear Regression” by Ravi Shastri (IJTSRD 2021)

[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html Scikit-learn - machine learning in python

[5] https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html Matplotlib - Pyplot in python